orgmode examples

draw, code evaluation and present in orgmode with LaTeX beamer

kimim

Document ID: Revision: Updated on: October 3, 2021

Outline

1. Introduction

2. Preparation

3. Drawing in code

4. Org-babel Evaluating Programming Languages

5. Org-babel for music and others

6. Presenting with Org-beamer

7. Conclusion

orgmode examples

1. Introduction

2. Preparation

3. Drawing in code

4. Org-babel Evaluating Programming Languages

5. Org-babel for music and others

6. Presenting with Org-beamer

7. Conclusion

Introduction

Purpose

To evaluate many features of orgmode, such as

- drawing with code
- evaluating results of code snippets
- exporting orgmode file to pdf slides

All the orgmode "source text" is hosted in github: `https://github.com/kimim/orgmode-examples`

Introduction

How

Following tools are used in this file:

- MSYS2 provides many tools and libraries
- GraalVM JDK supports Java, JS, R and more
- GNU Emacs with kimim-emacs configuration
- Org Mode, including org-babel, org-export
- TexLive with beamertheme-kimim style
- PlantUML, Graphviz, Mermaid, ditaa, LaTeX tikz package
- LilyPond for music notation
- Inkscape to convert svg to pdf, during orgmode-pdf exporting

orgmode examples

1. Introduction

2. Preparation

3. Drawing in code

4. Org-babel Evaluating Programming Languages

5. Org-babel for music and others

6. Presenting with Org-beamer

7. Conclusion

Preparation

Emacs settings

You may need to use kimim-emacs configuration:

```
# backup existing emacs config
cd ~ && mv .emacs .emacs-backup && mv .emacs.d .emacs.d-backup
# clone this config
git clone https://github.com/kimim/kimim-emacs
# copy default .emacs to ~
cp kimim-emacs/.emacs ~
```


Preparation

Emacs and Orgmode version

Firstly, let's check GNU Emacs[1] and Orgmode[2] version:

```
(concat  (emacs-version)
         "\nOrgmode " (org-version))
```

```
GNU Emacs 29.0.50 (build 1, x86_64-w64-mingw32)
 of 2021-10-03
Orgmode 9.5
```

[1] https://www.gnu.org/software/emacs

[2] https://orgmode.org

Preparation

TexLive and Beamer Theme

Install TexLive[3] to `<texlive-path>` and clone beamertheme-kimim[4], and update TEX cache:

```
git clone https://github.com/kimim/beamertheme-kimim \
    <texlive-path>/texmf-local/tex/latex/beamertheme-kimim
mktexlsr
```

[3] http://tug.org/texlive

[4] https://github.com/kimim/beamertheme-kimim

Preparation

Inkscape version

Install Inkscape[5] to convert SVG image to PDF.

This is inkscape version on my Windows 10:

```
inkscape --version
```

```
Inkscape 1.1.1 (3bf5ae0d25, 2021-09-20)
```

[5] https://inkscape.org

orgmode examples

1. Introduction

2. Preparation

3. Drawing in code

4. Org-babel Evaluating Programming Languages

5. Org-babel for music and others

6. Presenting with Org-beamer

7. Conclusion

Drawing in code

PlantUML settings in Emacs

Download plantuml.jar[6], and set jar-path

```
(require 'url-handlers)
(url-copy-file "https://nchc.dl.sourceforge.net/project/plantuml/plantuml.jar"
               "./plantuml.jar" t)

(use-package ob-plantuml
  :ensure nil
  :config
  (require 'plantuml-mode)
  ;; WARNING: if variables are from other package, setq them at :config
  (setq org-plantuml-jar-path "./plantuml.jar")
    (setq org-plantuml-executable-args "-headless -charset UTF-8")
    (add-to-list 'org-src-lang-modes '("plantuml" . plantuml))
    (add-to-list 'org-babel-load-languages '(plantuml . t)))
```

[6] https://plantuml.com

Drawing in code

PlantUML version

Here is the version info on my machine, including JVM, dot and graphviz:

```
(shell-command-to-string
 (concat
  "java -jar " org-plantuml-jar-path " -version"))
```

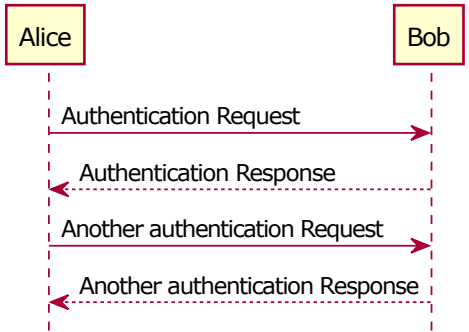
```
PlantUML version 1.2021.8 (Sat Jun 26 16:20:59 CST 2021)
(GPL source distribution)
Java Runtime: OpenJDK Runtime Environment
JVM: OpenJDK 64-Bit Server VM
Default Encoding: GBK
Language: en
Country: US

PLANTUML_LIMIT_SIZE: 4096

Dot version: dot - graphviz version 2.44.1 (20200629.0846)
Installation seems OK. File generation OK
```


Drawing in code

Sequence Diagram

Let's draw a simple sequence diagram with this plantuml code:

```
@startuml
hide footbox
hide unlinked
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response
Alice -> Bob: Another authentication Request
Alice <-- Bob: Another authentication Response
@enduml
```

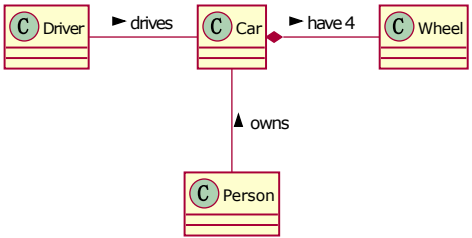


Drawing in code

Class Diagram

A simple class diagram

```
@startuml
class Car
Driver - Car : drives >
Car *- Wheel : have 4 >
Car -- Person : < owns
@enduml
```



Drawing in code

ditaa settings in Emacs

Download ditaa.jar[7], and set jar-path

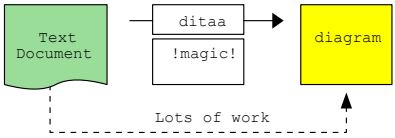
```
(require 'url-handlers)
(url-copy-file "https://sourceforge.net/projects/ditaa/files/latest/download"
               "./ditaa.zip" t)
(shell-command-to-string "unzip ditaa.zip && mv ditaa*.jar ditaa.jar")

(use-package ob-ditaa
  :ensure nil
  :custom
  (org-ditaa-jar-path "./ditaa.jar")
  :config
  (add-to-list 'org-src-lang-modes '("ditaa" . artist))
  (add-to-list 'org-babel-load-languages '(ditaa . t)))
```

[7] https://sourceforge.net/projects/ditaa/files/latest/download

Drawing in code

ditaa

```
+--------+   +-------+    +-------+
|cGRE    | --+ ditaa +--> |cFF0   |
|  Text  |   +-------+    |diagram|
|Document|   |!magic!|    |       |
|     {d}|   |       |    |       |
+---+----+   +-------+    +-------+
    :                         ^
    |       Lots of work      |
    +-------------------------+
```



Drawing in code

mermaid settings in Emacs

Install mermaid with npm:

```
npm install -g @mermaid-js/mermaid-cli
```

Configure mermaid in emacs

```
(use-package ob-mermaid
  :custom
  (ob-mermaid-cli-path "~/node_modules/.bin/mmdc.cmd")
  :config
  (add-to-list 'org-babel-load-languages '(mermaid . t)))
```


Drawing in code

mermaid version

```
(shell-command-to-string
 (concat ob-mermaid-cli-path " --version"))
```

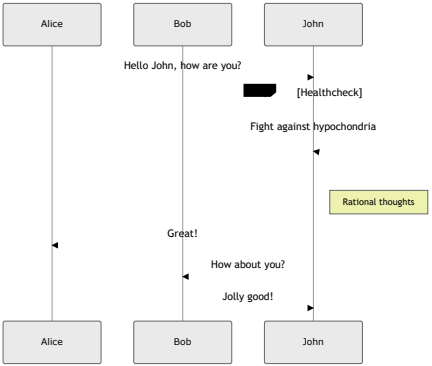
```
8.12.1
```


Drawing in code

mermaid

Installation and Configuration, see
kimim-emacs#mermaid

```
sequenceDiagram
    participant Alice
    participant Bob
    Alice->>John: Hello John, how are you?
    loop Healthcheck
        John->>John: Fight against hypochondria
    end
    Note right of John: Rational thoughts
    John-->>Alice: Great!
    John->>Bob: How about you?
    Bob-->>John: Jolly good!
```



Drawing in code

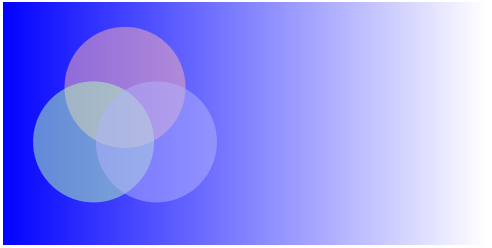
tikz logo

```
\begin{tikzpicture}
  \filldraw[blue] (0,0) rectangle (-4,-2);
  \filldraw[blue,rotate=-30] (0,0) rectangle (1,-0.2);
  \filldraw[blue] (-4,0) circle (0.2);
  \filldraw[blue] (-4,-2) rectangle (-3,-3);
  \filldraw[blue] (0,-2) rectangle (-1,-3);
  \filldraw[blue] (-4,0) rectangle (-5.5,-1.5);
\end{tikzpicture}
```

Drawing in code

tikz background

```
\begin{tikzpicture}
  \fill[left color=blue,right color=white,shading angle=90,line width=0] (0,0) rectangle (4,-2);
  \begin{scope}[shift={(1,-1)}]
    \fill[red!30!white,semitransparent]   ( 90:0.3) circle (0.5);
    \fill[green!30!white,semitransparent] (210:0.3) circle (0.5);
    \fill[blue!30!white,semitransparent]  (330:0.3) circle (0.5);
  \end{scope}
\end{tikzpicture}
```



orgmode examples

1. Introduction

2. Preparation

3. Drawing in code

4. Org-babel Evaluating Programming Languages

5. Org-babel for music and others

6. Presenting with Org-beamer

7. Conclusion

Org-babel Evaluating Programming Languages

emacs lisp

```
(emacs-version)
```

```
GNU Emacs 28.0.50 (build 6, x86_64-w64-mingw32)
 of 2021-08-31
```

```
(decoded-time-year (decode-time (current-time)))
```

```
2021
```


Org-babel Evaluating Programming Languages

shell

```
sh --version
```

```
GNU bash, version 5.1.8(1)-release (x86_64-pc-msys)
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```


Org-babel Evaluating Programming Languages

C

```c
printf("%s is %d years old\n", "C programming language", year-1972);
```

```
C programming language is 49 years old
```


Org-babel Evaluating Programming Languages

C++

```
cout << "C++ is " << year-1979 << " years old" << endl;
```

```
C++ is 42 years old
```


Org-babel Evaluating Programming Languages

Clojure

```
(println "Clojure is" (- year 2005) "years old")
```

```
Clojure is 16 years old
```


Org-babel Evaluating Programming Languages

ClojureScript

TODO

Org-babel Evaluating Programming Languages

Java

TODO: can pass variable to java

```
public class Main{
    public static void main(String[] args){
        System.out.println("Java is " + (2021-1995) + " years old");
        return;
    }
}
```

```
Java is 26 years old
```


Org-babel Evaluating Programming Languages

Python

Check Python version in shell:

```
python --version
```

```
Python 3.9.6
```

Evaluate Python code:

```
print("Python is " + str(year - 1991) + " years old")
```

```
Python is 30 years old
```


Org-babel Evaluating Programming Languages

Rust

```
(package-install 'ob-rust)
```

TODO: cannot pass variable to rust

```
fn main() {
    println!("Rust is {} years old", 2021 - 2016);
}
```


Org-babel Evaluating Programming Languages

Go

TODO

```
package main
import ("fmt")

func main(){
   fmt.Println("emacs")
}
```


Org-babel Evaluating Programming Languages

R

TODO

orgmode examples

1. Introduction

2. Preparation

3. Drawing in code

4. Org-babel Evaluating Programming Languages

5. Org-babel for music and others

6. Presenting with Org-beamer

7. Conclusion

Org-babel for music and others

Install LilyPond

Get the installation file from https://lilypond.org/download.html and install it.

Check the version:

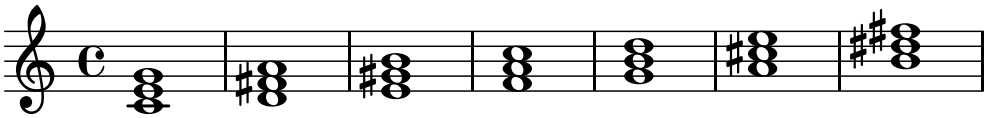
```
lilypond --version
```

```
GNU LilyPond 2.22.1

Copyright (c) 1996--2021 by
  Han-Wen Nienhuys <hanwen@xs4all.nl>
  Jan Nieuwenhuizen <janneke@gnu.org>
  and others.

This program is free software.  It is covered by the GNU General Public
License and you are welcome to change it and/or distribute copies of it
under certain conditions.  Invoke as `lilypond --warranty' for more
information.
```


Org-babel for music and others

LilyPond

```
\relative c' {
  \chordmode {c1}
  \chordmode {d1}
  \chordmode {e1}
  \chordmode {f1}
  \chordmode {g1}
  \chordmode {a1}
  \chordmode {b1}
}
```



orgmode examples

1. Introduction

2. Preparation

3. Drawing in code

4. Org-babel Evaluating Programming Languages

5. Org-babel for music and others

6. Presenting with Org-beamer

7. Conclusion

Presenting with Org-beamer

Beamer

In this section, I will try some beamer settings in orgmode.

Presenting with Org-beamer

latexmk version

```
(princ (concat (format "LaTeXmk version: %s\n"
                       (eshell-command-result "latexmk --version") "\n")
               (format "XeTeX version: %s\n"
                       (eshell-command-result "xelatex --version") "\n")))
```

```
LaTeXmk version: Latexmk, John Collins, 29 May 2021. Version 4.74b
XeTeX version: XeTeX 3.141592653-2.6-0.999993 (TeX Live 2021/W32TeX)
kpathsea version 6.3.3
Copyright 2021 SIL International, Jonathan Kew and Khaled Hosny.
There is NO warranty.  Redistribution of this software is
covered by the terms of both the XeTeX copyright and
the Lesser GNU General Public License.
For more information about these matters, see the file
named COPYING and the XeTeX source.
Primary author of XeTeX: Jonathan Kew.
Compiled with ICU version 68.2; using 68.2
Compiled with zlib version 1.2.11; using 1.2.11
Compiled with FreeType2 version 2.10.4; using 2.10.4
Compiled with Graphite2 version 1.3.14; using 1.3.14
Compiled with HarfBuzz version 2.7.4; using 2.7.4
Compiled with libpng version 1.6.37; using 1.6.37
Compiled with pplib version v2.05 less toxic i hope
Compiled with fontconfig version 2.13.93; using 2.13.93
```


Presenting with Org-beamer

simple slide

This is a simple slide, with some formatted texts:

- important underline *slashed* `code` `verbatim` ~~deleted~~ alert
  - important underline *slashed* `code` `verbatim` ~~deleted~~ alert
  - important underline *slashed* `code` `verbatim` ~~deleted~~ alert
    - important underline *slashed* `code` `verbatim` ~~deleted~~ alert
    - important underline *slashed* `code` `verbatim` ~~deleted~~ alert
    - important underline *slashed* `code` `verbatim` ~~deleted~~ alert

Enumerations:

1. important underline *slashed* `code` `verbatim` ~~deleted~~ alert

1.1 important underline *slashed* `code` `verbatim` ~~deleted~~ alert

1.2 important underline *slashed* `code` `verbatim` ~~deleted~~ alert

1.2.1 important underline *slashed* `code` `verbatim` ~~deleted~~ alert

1.2.2 important underline *slashed* `code` `verbatim` ~~deleted~~ alert

1.2.3 important underline *slashed* `code` `verbatim` ~~deleted~~ alert

Presenting with Org-beamer

simple slide with definition

It is not recommended to have second level definition bullet…

- **Beamer** LaTeX package to generate slides
- **Orgmode** Powerful plain text format
  - **org-babel** Let Orgmode understand and evaluate programming languages
  - **ox-latex** Exporter to export orgmode to latex and further to PDF

Presenting with Org-beamer

simple slide with wallpaper

- This slide has a nice wallpaper.
- It is the westlake in the morning.

Presenting with Org-beamer

special blocks with heading - 1

block

- this is a block

alert block

- this is an alert block

Theorem (theorem block)

- *this is a theorem block*

proof.

- This is proof

□

Presenting with Org-beamer

special blocks with heading - 2

Example (example)

This is an example

example block

Example block

Definition (definition)

- this is a definition

Presenting with Org-beamer

special blocks without heading

- this is a beamercolorbox

  $verse\ is\ a\ poem?\ maybe.$

> *Software is eating the world.*

> *This is a quote.*

Presenting with Org-beamer

some todo list

- daily task [33%]
  - ☒ fetch the milk in the morning
  - ☐ check the mailbox
  - ☐ clean the garden
- learning task [50%]
  - ☒ read the book
  - ☒ write the reading notes
  - ☐ make a presentation
  - ☐ present to students

Presenting with Org-beamer

table

enrollment to the class	name	date
x	Kimi	2021-09-18
	Ivy	2021-09-28
x	Anna	2021-09-20

Presenting with Org-beamer

4 dimension

up-left

- 1

up-right

- 2

down-left

- 3

down-right

- 4

Presenting with Org-beamer

three columns

col1

- left column occupies 33%

col2

- middle column occupies 33%

col3

- right column occupies 33%

Presenting with Org-beamer

quote and quotation

Quote:

> *If winter comes, can Spring be far behind?*

Quotation:

> *History repeats itself, and that's one of the things that's wrong with history.*

Presenting with Org-beamer

Highlighting text

The double `@@` can be used to enclose active code.

For example:

```
This is very *@@beamer:<2->@@important*
```

This is very important

Presenting with Org-beamer

Highlighting text

The double `@@` can be used to enclose active code.

For example:

```
This is very *@@beamer:<2->@@important*
```

This is very important

Presenting with Org-beamer

Lists in action

`#+ATTR_BEAMER: :overlay +-` can show list one by one:

- Can you remember

You can also add `<3->` before each item to set the order:

Presenting with Org-beamer

Lists in action

`#+ATTR_BEAMER: :overlay +-` can show list one by one:

- Can you remember
- The magic Mesosoic numbers

You can also add `<3->` before each item to set the order:

Presenting with Org-beamer

Lists in action

`#+ATTR_BEAMER: :overlay +-` can show list one by one:

- Can you remember
- The magic Mesosoic numbers
- The dinosaur one to ten

You can also add `<3->` before each item to set the order:

Presenting with Org-beamer

Lists in action

`#+ATTR_BEAMER: :overlay +-` can show list one by one:

- Can you remember
- The magic Mesosoic numbers
- The dinosaur one to ten

You can also add `<3->` before each item to set the order:

- The dinosaur one to ten

Presenting with Org-beamer

Lists in action

`#+ATTR_BEAMER: :overlay +-` can show list one by one:

- Can you remember
- The magic Mesosoic numbers
- The dinosaur one to ten

You can also add `<3->` before each item to set the order:

- The magic Mesosoic numbers
- The dinosaur one to ten

Presenting with Org-beamer

Lists in action

`#+ATTR_BEAMER: :overlay +-` can show list one by one:

- Can you remember
- The magic Mesosoic numbers
- The dinosaur one to ten

You can also add <3-> before each item to set the order:

- Can you remember
- The magic Mesosoic numbers
- The dinosaur one to ten

Presenting with Org-beamer

Columns in action

Phoenix! Phoenix! How virtue has declined.
It can't wait for the future or catch up with what's behind.
When the Dao works in the world, the sage man works his ways,
When the Dao has disappeared the Sage lives out his days.

In times like these just keep far from the shackles and the blade.
Good fortune's lighter than a feather, but none knows how to bear it,
Disaster's heavier than the earth, but none knows how to dodge it.

Enough! Enough! These toils of virtue serving man,
Danger! Danger! Escape! –draw the line in the sand.
Brambles, brambles, don't cut me as I go,
Twisting, twisting, my feet stay free of woe.

Presenting with Org-beamer

Columns in action

Phoenix! Phoenix! How virtue has declined.
It can't wait for the future or catch up with what's behind.
When the Dao works in the world, the sage man works his ways,
When the Dao has disappeared the Sage lives out his days.

In times like these just keep far from the shackles and the blade.
Good fortune's lighter than a feather, but none knows how to bear it,
Disaster's heavier than the earth, but none knows how to dodge it.

Enough! Enough! These toils of virtue serving man,
Danger! Danger! Escape! –draw the line in the sand.
Brambles, brambles, don't cut me as I go,
Twisting, twisting, my feet stay free of woe.

Presenting with Org-beamer

Columns in action

Phoenix! Phoenix! How virtue has declined.
It can't wait for the future or catch up with what's behind.
When the Dao works in the world, the sage man works his ways,
When the Dao has disappeared the Sage lives out his days.

In times like these just keep far from the shackles and the blade.
Good fortune's lighter than a feather, but none knows how to bear it,
Disaster's heavier than the earth, but none knows how to dodge it.

Enough! Enough! These toils of virtue serving man,
Danger! Danger! Escape! –draw the line in the sand.
Brambles, brambles, don't cut me as I go,
Twisting, twisting, my feet stay free of woe.

Presenting with Org-beamer

Columns in action

Phoenix! Phoenix! How virtue has declined.
It can't wait for the future or catch up with what's behind.
When the Dao works in the world, the sage man works his ways,
When the Dao has disappeared the Sage lives out his days.

In times like these just keep far from the shackles and the blade.
Good fortune's lighter than a feather, but none knows how to bear it,
Disaster's heavier than the earth, but none knows how to dodge it.

Enough! Enough! These toils of virtue serving man,
Danger! Danger! Escape! –draw the line in the sand.
Brambles, brambles, don't cut me as I go,
Twisting, twisting, my feet stay free of woe.

Presenting with Org-beamer

Blocks in action

Block1

Great understanding is broad, small understanding is picky.

Presenting with Org-beamer

Blocks in action

Block1

Great understanding is broad, small understanding is picky.

Block2

Great words overflowing, small words haggling.

Presenting with Org-beamer

Blocks in action

Block1

Great understanding is broad, small understanding is picky.

Block2

Great words overflowing, small words haggling.

Block3

Asleep the bodily soul goes roaming, awake it opens through our form.

orgmode examples

1. Introduction

2. Preparation

3. Drawing in code

4. Org-babel Evaluating Programming Languages

5. Org-babel for music and others

6. Presenting with Org-beamer

7. Conclusion

Conclusion

Key Takeaways

- Emacs is a long lasting, and wonderful text editor
- Orgmode is an awesome plain text format
- LaTeX is great typesetting tool
- Beamer is a LaTeX package for preparing presentation
- Thus, using these tools within Emacs is cool!

Appendix

References I